Machine Learning Model of Phase Design for High Entropy Alloys

ZHOU Ziqing

Copyright 2018 The MathWorks, Inc.

This seminar contains two major parts:

Part1 – Alloy features calculation

Transforming the alloy components and fractions into machine learning features.

Part2 – Artificial neural network (ANN)

Applying ANN to the phase design problem

**Part 1: Alloy features calculation**

The goal is to transform the alloy components and fractions into 13 different features which will be applied as the input of the machine learning models (including artificial neural network, convolutional neural network and support vector machine). The 13 features are shown in the following **Table 1**.

**Table 1**. The thirteen design parameters and the corresponding formula. Note that *ci* stands for the atomic fraction of the *ith* element.

|  |  |
| --- | --- |
| Parameters | Formula |
| mean atom radius |  |
| atomic size difference |  |
| Average melting temperature |  |
| Standard deviation of melting temperature |  |
| Average mixing enthalpy |  |
| Standard deviation of mixing enthalpy |  |
| Ideal mixing entropy |  |
| Electronegativity |  |
| Standard deviation of electronegativity |  |
| Average VEC |  |
| Standard deviation of VEC |  |
| Mean bulk modulus |  |
| Standard deviation of bulk modulus |  |

This part shows:

a) The dataset of 601 alloys with the detailed components and fractions, labeled by SS, IM, and AM.

b) The prediction dataset with the detailed components and fractions.

c) The basic properties of elements

d) The mixing enthalpy between 2 elements

e) The program to calculate features

**Part 2: Artificial neural network**

This part aims to build a machine learning model with high prediction accuracy, using the calculated 13 features of 601 alloys dataset. Since the label AM, IM and SS are not independent, the models were trained with the label of 0 and 1, where 0 means the absence of the phase (AM, IM or SS) while 1 means the existence. Thus, three models are trained, including the AM, IM and SS. Then a sensitivity matrix was extracted from the ANN model to value the feature importance, which helps to do the feature reduction. The testing accuracy is measured to be the criterion of the performance. After the feature reduction, the model with the best performance is picked up to do the prediction.

This part shows:

a) The dataset with calculated 13 features

b) One layer feed-forward neural network with 20 hidden layers

c) The training results of ANN models, with different number of features

d) The sensitivity matrix extracted from the ANN models

e) The well-trained models

f) The prediction dataset with calculated 13 features and the prediction results

**Steps to run the seminar:**

To run the seminar, please follow the steps below. The seminar is running in the environment R2018b.

Required products:

MATLAB, Neural Network Toolbox, MATLAB Compiler

**Part1: Alloy features calculation**

1. Running “featurescalculate.m” from \calculatefeatures
2. Copying the variable “features” (with a size of 601x13) in the workspace
3. Opening the excel file “dataset.xlsx” and pasting the variable “features” to E2:Q602

**Part2: Artificial neural network**

1. Copying the excel file “dataset.xlsx” to \ANNmodel.
2. Training the ANN model. Running from MATLAB: Run “NNSTART.m” from \ANNmodel.

In the dialogue box of “Neural Network (nnstart)”, following the steps:

a. Clicking “Pattern Recognition app”

b. In the dialogue box of “Neural Pattern Recognition”, clicking “Next”

c. Selecting “features” as “Inputs” and “AM”, “IM”, or “SS” as Targets

d. Selecting “Matrix rows” and clicking “Next”

e. Clicking “Next”

f. Changing the “Number of Hidden Neuros” to 20; clicking “Next”

g. Clicking “train” and starting training the neural network

h. In the dialogue box of “Neural Network Training”, finding the progress and result.

1. Extracting the Sensitivity Matrix:

a. In the dialogue box of “Neural Pattern Recognition”, clicking “Next” twice and clicking “Matlab function” to get the function of well-trained neural network.

b. Writing the variables “IW1\_1” and “LW2\_1”in the created function to the workspace.

c. In the command window, typing in “LW2\_1\*IW1\_1” and the result is the sensitivity matrix.

d. Copying the sensitivity matrix to the excel file in \Sensitivitymatrix.

1. Saving the well-trained function in \ANNresults and changing the function name with a formation of “Phase + No. of features”. For example, AM13 means the AM phase model with 13 training features.
2. Training models with different labels: repeating the steps 1, 2, 3 and 4 by selecting different Targets ( AM, IM or SS)
3. Training models with reduced features: running “FeatureReduction\_NNSTART.m”, repeating steps 1-5 by selecting different “features\_XX” (XX represents the number of features)
4. Predicting new phases:

a. Selecting the function with the best performance (“AM9”, “IM11” and “SS10”);

b. Running “NNprediction.m”

c. Copying the variable “y” in the workspace to the excel file “test.xlsx” S2:U9

d. Rounding the data in S2:U9 to 0 or 1; 0 means absence of the phase while 1 means existence of the phase.